In [54]:

```python
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from pylab import *
```
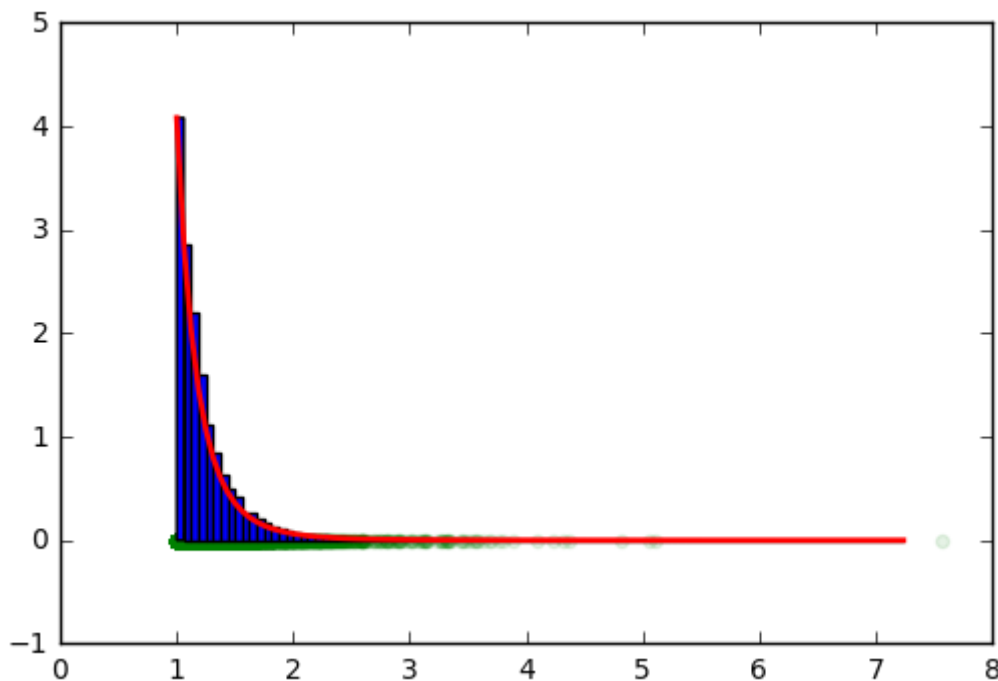
In [55]:

```python
# Рапсределение Парето с параметром 5
a, m = 5, 1
data = (np.random.pareto(a, 10000) + 1) * m

import matplotlib.pyplot as plt
count, bins, _ = plt.hist(data, 100, normed=True)
fit = a*m**a / bins**(a+1)
plt.plot(bins, max(count)*fit/max(fit), linewidth=2, color='r')
scatter(s, np.zeros(10000), color = "green", alpha=0.1)
plt.show()
```
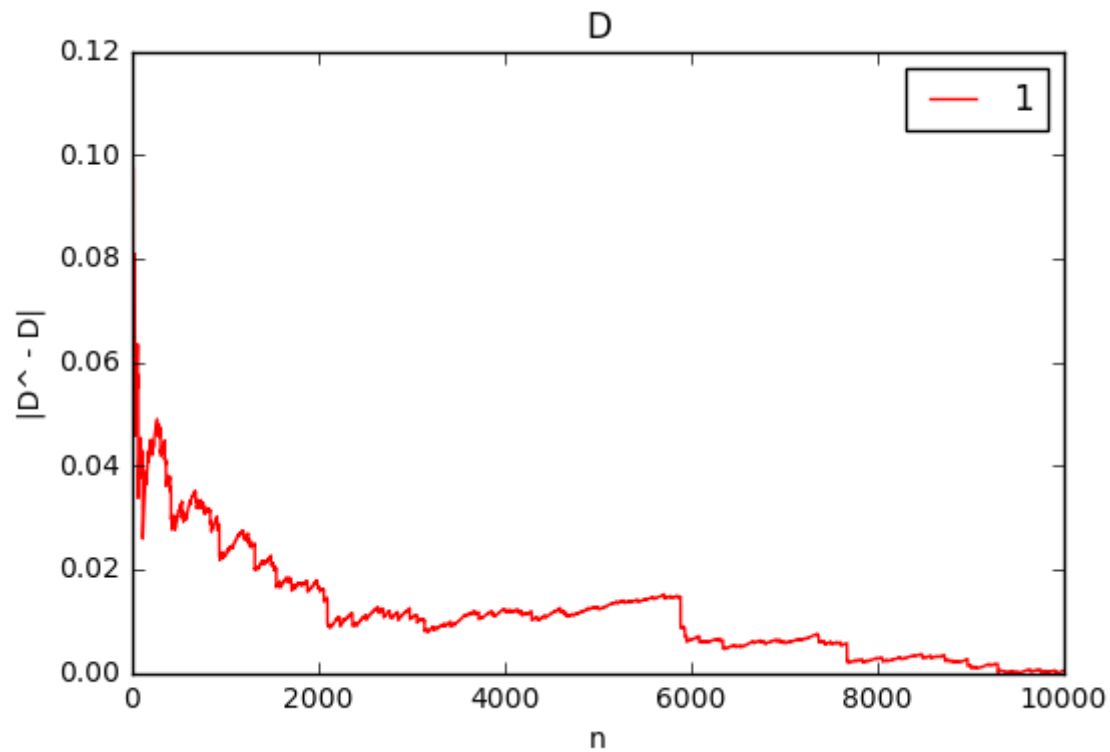

In [56]:

```python
# Массив квадратов
arr_sqr = []
for i in data:
    arr_sqr += [i ** 2]
```

In [67]:

```python
# Заполнение массива разности выборочной дисперсии с истинной
s = []
for i in range(1, N+1):
    s += [abs(np.array(arr_sqr[:i]).sum() / i - (np.array(data[:i]).sum() / i) ** 2
```

In [68]:

```
plt.plot(range(1, N + 1), s, color = 'red', label='1')
plt.legend(loc='best')
plt.title('D')
plt.xlabel('n')
plt.ylabel('|D^ - D|');
plt.show()
```
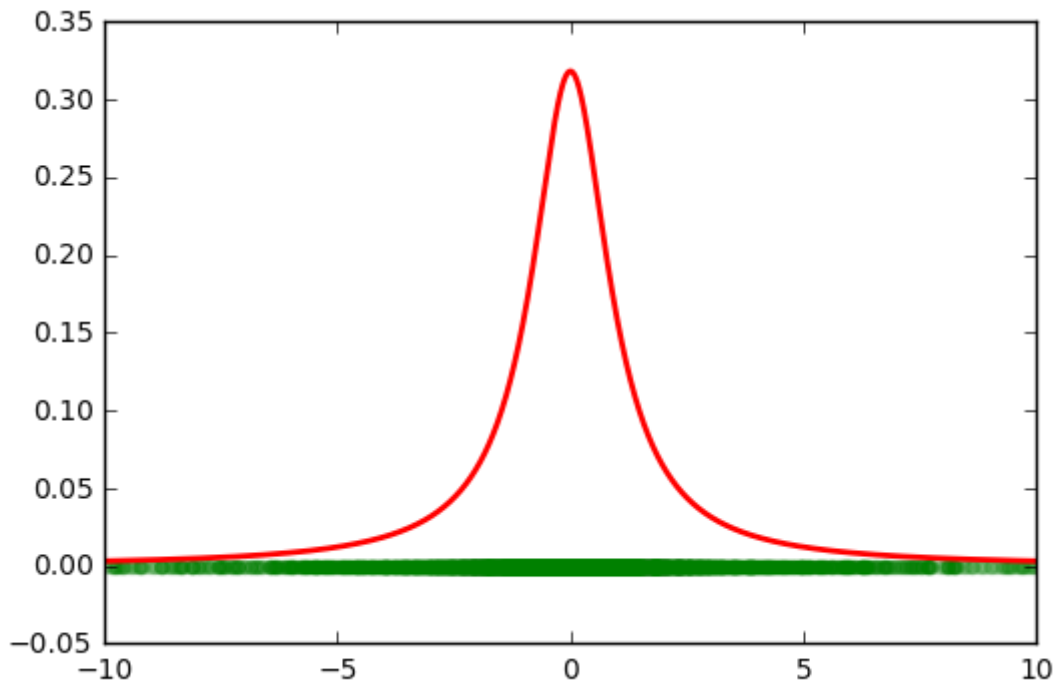
In [79]:

```python
# Стандартное рапспределение Коши
data = (np.random.standard_cauchy(size=N))
def cauch(x):
    return 1.0 / (3.141592 * (1 + x **2) )

xlim(-10,10)
bins = np.linspace(-10, 10, 4000)
arr = []
for i in bins:
    arr += [cauch(i)]
plt.plot(bins, arr, linewidth=2, color='r')
scatter(data, np.zeros(N), color = "green", alpha=0.1)
plt.show()
```


In [80]:

```python
# Массив квадратов
arr_sqr = []
for i in data:
    arr_sqr += [i ** 2]
```

In [81]:

```python
# Заполнение массива  выборочной дисперсии
s = []
for i in range(1, N+1):
    s += [np.array(arr_sqr[:i]).sum() / i - (np.array(data[:i]).sum() / i) ** 2 ]
```

In [82]:

```python
plt.plot(range(1, N + 1), s, color = 'red', label='1')
plt.legend(loc='best')
plt.title('D')
plt.xlabel('n')
plt.ylabel('D^');
plt.show()
```